一次元2近傍可逆的セル・オートマトンについて(アルゴリズムと計算量の理論)

森田, 憲一

数理解析研究所講究録
731巻 1990年 108-117

1次元2近傍可逆的セル・オートマトンについて

山形大学工学部　　森田 憲一 (Kenichi Morita)

要旨　可逆的（単射）セル・オートマトンとは、その各々の状相が1時刻前の状相を高々1つしか持たないような、つまり、一意的に時間を逆戻りできるようなセル・オートマトン（CA）である。著者らはこれまでに、1次元3近傍可逆的CAが計算万能であること、つまり、任意のTuring機械を模倣できることを示している。本稿では、1次元2近傍可逆的CAについて考察する。ここでは、1次元 Partitioned セル・オートマトン（PCA）と呼ぶ1次元CAのあるサブクラスを考え、任意の3近傍可逆的PCAが2近傍可逆的PCAによって模倣できることを示す。これと3近傍可逆的PCAが計算万能であるという結果から、2近傍可逆的PCA（およびCA）の計算万能性が導かれる。

1.　まえがき

可逆的なオートマトンのシステムとは、そのシステムの各時刻における全状態（計算状況）から直前の時刻の全状態が常に唯一に定まる、つまり逆方向の状態遷移が決定的であるようなものをいう。このような性質（論理的な可逆性）は一見非常に特殊な性質に見えるかもしれないが、実際には物理的あるいは化学的な可逆性と密接な関係があり、システムにおける不可避的なエネルギー消費の問題を理論的に論じる上で重要な意味を持つことが知られている[2],[4]。また、エネルギーの問題から離れたとしても、可逆性という特別な性質を持つシステムがどのような振る舞いをするか（あるいは計算能力を持つか）という問題自体に興味深い点がある。（可逆的システムの研究の歴史は文献(3)を参照。）

本論文の研究対象は可逆的なセル・オートマトンであるが、その前に、可逆的Turing機械について述べておく。可逆的Turing機械を最初に定式化したのは、Bennettである[1]。Bennettは任意の（一般には非可逆な）Turing機械が、可逆的

な３テープ多記号Turing機械によって模倣できることを示し、可逆性という制約がTuring機械の計算能力に対する制限にはならないことを明らかにした。ここで注意すべきことは、Turing機械を可逆化するには、各ステップにおいてどの遷移規則が使われたかを特別の「履歴テープ」に逐一記録しておけばよく、ある意味で非常に簡単にできることである。しかしこれだけだと計算終了時に履歴情報が大量のゴミ情報として残されてしまい、問題解決を遅らせたにすぎない（ゴミ情報の単純な消去は非可逆的な過程となる）。Bennettの構成法で重要なのは、可逆性を保ったままでゴミ情報を最終的にきれいに消去できるという点である。また森田ら[5]は、Bennettの方法を改良して、任意のTuring機械から、それを模倣する１テープ２記号の可逆的Turing機械を構成する具体的方法を与えた。

セル・オートマトン（ＣＡ）は、同一の有限オートマトンが空間的に一様に配列、接続されたシステムであるが、このようなシステムの可逆性は、大域写像（状相から状相への写像）の単射性として定義される。可逆的ＣＡの研究としては、大域写像の単射性と全射性の関係を研究したものもあるが、計算能力の研究にはToffoli[7]のものがある。Toffoliはｋ次元の非可逆なＣＡがｋ＋１次元の可逆的ＣＡによって模倣できることを示し、これにより２次元可逆的ＣＡが計算万能であるという結果を得ている。一方、森田ら[6]はこの結果を強め、１次元３近傍可逆的ＣＡが計算万能であるという結果を与えている。後者の結果は、可逆的な１テープTuring機械を１次元３近傍可逆的ＣＡで模倣することによって得られるが、その際、Partitioned セル・オートマトン（ＰＣＡ）と呼ぶ、ＣＡのあるサブクラスを新しく導入して証明がなされている。ＰＣＡは、１つのセルがいくつかの部分に分割されたようなＣＡで、大域写像の単射性（つまり可逆性）と局所写像の単射性とが等価になるようなシステムである。それゆえ、１テープ可逆的Turing機械を模倣する可逆的ＰＣＡが容易に設計できるという利点がある。

本稿では、２近傍のＰＣＡ（２ＰＣＡ）について考察する。ここで考える２ＰＣＡは、着目するセルの次の時刻の状態が自身と左隣のセルの状態だけによって決まる、つまり、情報が１方向にしか流れないＣＡである。ここではまず、任意の３近傍ＰＣＡ（３ＰＣＡ）が２ＰＣＡによって模倣できること、また特に、任意の可逆的３ＰＣＡが可逆的２ＰＣＡによって模倣できることを示す。可逆的３ＰＣＡの計算万能性は既に得られているので、これにより、可逆的２ＰＣＡ（および２近傍の可逆的ＣＡ）が計算万能であるという結果が導かれる。この模倣法では、模倣される側が静止状態付きのＰＣＡである場合、模倣するＰＣＡも静止状態付きのものとして構成される。

2.　諸定義と準備

決定性１次元セル・オートマトン（ＣＡ）とは、

$$A = (Z,\ Q,\ N,\ f,\ q_0)$$

によって定義されるシステムである。但し、各項は以下の通り。

(1) Ｚはすべての整数の集合。

(2) Ｑは各セルの内部状態の空でない有限集合。

(3) $N = (n_1, n_2, \cdots, n_k)$ は近傍（各n_iはＺの要素）。

(4) $f : Q^k \to Q$ は各セルの状態遷移を定める局所写像。

(5) $q_0 \in Q$ は静止状態（$f(q_0, q_0, \cdots, q_0) = q_0$ を満たす）。

ＣＡの内で特に、$N = (-1, 0, +1)$ であるようなＣＡを３近傍セル・オートマトン（３ＣＡ）、$N = (-1, 0)$ であるようなＣＡを２近傍セル・オートマトン（２ＣＡ）あるいは１方向セル・オートマトンと呼ぶ。

集合Ｑ上の（Ａの）状相とは、

$$c :\ Z \to Q$$

なる写像ｃのことである。Ｑ上のすべての状相の集合を Conf(Q) で表す。つまり、

$$\mathrm{Conf}(Q) = \{c \mid c : Z \to Q\}$$

である。ｆによって定まる次のような写像 $F : \mathrm{Conf}(Q) \to \mathrm{Conf}(Q)$ を大域写像（並列写像）と呼ぶ。

$$F(c)(i) = f(c(i+n_1), c(i+n_2), \cdots, c(i+n_k))$$

Ａが可逆的であるとは、大域写像Ｆが単射であることをいう。

決定性１次元３近傍 Partitioned セル・オートマトン（３ＰＣＡ）[6]とは、

$$P = (Z,\ (L,\ C,\ R),\ g,\ (l_0, c_0, r_0))$$

によって定義されるシステムである。但し、各項は以下の通り。

(1) Ｚはすべての整数の集合。

(2) Ｌ，Ｃ，Ｒはそれぞれ、左状態、中央状態、右状態と呼ばれる、状態の空でない有限集合で、Ｌ×Ｃ×Ｒの元が１個のセルの状態を表す。

(3) $g : R \times C \times L \to L \times C \times R$ は各セルの状態遷移を定める局所写像。

(4) $(l_0, c_0, r_0) \in L \times C \times R$ は静止状態。
　（$g(r_0, c_0, l_0) = (l_0, c_0, r_0)$を満たす。）

３ＰＣＡは、直観的には図２．１のように、１つのセルが３つの部分に分割されており、各セルの次の時刻の状態が、左隣のセルの右状態、そのセルの中央状

態、右隣のセルの左状態だけに依存して決まるようなＣＡである。

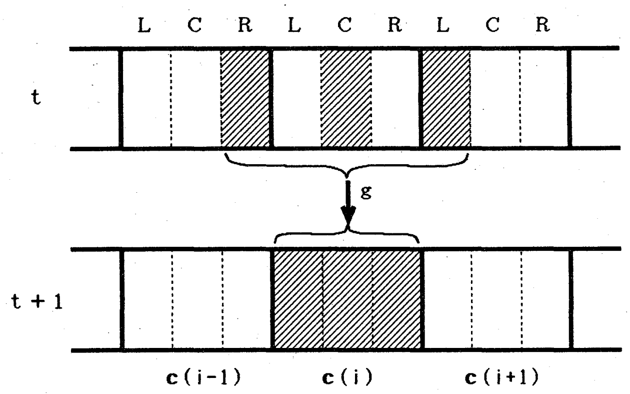

図２．１　３ＰＣＡの状態遷移

集合Ｌ×Ｃ×Ｒ上の（Ｐの）状相とは、

$$c: Z \to L \times C \times R$$

なる写像ｃのことである。Ｌ×Ｃ×Ｒ上のすべての状相の集合をＣＡの場合と同様 Conf(Ｌ×Ｃ×Ｒ）で表す。つまり、

$$\mathrm{Conf}(L \times C \times R) = \{ c \mid c : Z \to L \times C \times R \}$$

である。Ｌ×Ｃ×Ｒに属する３項組から、左、中央、右、の要素を取り出す射影関数をそれぞれ、“LEFT”，“CENTER”，“RIGHT”とする。このとき、Ｐによって定まる次のような写像 $G: \mathrm{Conf}(L \times C \times R) \to \mathrm{Conf}(L \times C \times R)$ を大域写像と呼ぶ。

$$G(c)(i) = g(\mathrm{RIGHT}(c(i-1)), \mathrm{CENTER}(c(i)), \mathrm{LEFT}(c(i+1)))$$

ＣＡの場合とは異なり、ＰＣＡに対しては大域可逆性と局所可逆性の２種類の可逆性が定義できる。つまり、Ｐが<u>大域可逆的</u>であるとは大域写像Ｇが単射であることをいい、Ｐが<u>局所可逆的</u>であるとは局所写像ｇが単射であることをいう。ＣＡの局所写像は $f: Q^k \to Q$ であるため、つまらない場合を除いて単射にはなり得ないが、３ＰＣＡの局所写像は $g: R \times C \times L \to L \times C \times R$ であるので単射になり得る。

３ＰＣＡの内で、集合Ｌの要素数が１個であるようなものを特に、決定性１次元２近傍（あるいは１方向）Partitioned セル・オートマトン（２ＰＣＡ）と呼ぶが、上の３ＰＣＡの定式化中のＬを省いて次のようなシステムとして定義することができる。

$$P = (Z, (C, R), g, (c_0, r_0))$$

但し、局所写像ｇは

$g : R \times C \rightarrow C \times R$

であり、大域写像Gはgから次のように定められる。

$G(c)(i) = g(\mathrm{RIGHT}(c(i-1)), \mathrm{CENTER}(c(i)))$

直観的には２ＰＣＡは、図２．２のように状態遷移をするシステムである。

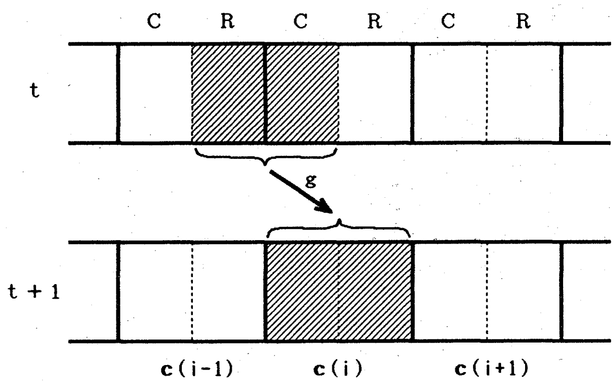

図２．２　２ＰＣＡの状態遷移

ＰＣＡやＣＡに関してはこれまでに以下のような結果が得られている。

【命題２．１】(6)　$P = (Z, (L, C, R), g, (l_0, c_0, r_0))$ を任意の３ＰＣＡとする。Ｐは、大域可逆的であるとき、かつそのときに限り、局所可逆的である。

この命題は、２ＰＣＡに対しても成り立つ（２ＰＣＡはＬの要素数が１であるような特別な３ＰＣＡであるので）。この命題により、以下ではＰＣＡが大域可逆的あるいは局所可逆的であることを単に可逆的と呼ぶことにする。

【命題２．２】(6) 任意の３ＰＣＡ, $P = (Z, (L, C, R), g, (l_0, c_0, r_0))$ に対し、大域写像がそれと一致するような（つまり、$F = G$となるような）３ＣＡ, $A = (Z, Q, (-1, 0, +1), f, q_0)$ が存在する。

これと同様の命題が２ＰＣＡと２ＣＡの間にも成り立つ。証明も同様であるので結果だけを次に示す。

【命題２．３】　任意の２ＰＣＡ, $P = (Z, (C, R), g, (c_0, r_0))$ に対し、大域写像がそれと一致するような（つまり、$F = G$となるような）２ＣＡ, $A = (Z, Q, (-1, 0), f, q_0)$ が存在する。

一方、可逆的Turing機械に関しては以下のように、Bennettの結果およびそれを改良した結果が導かれている。

【命題2．4】(1)　任意の（一般に非可逆な）Turing機械Tに対し、それを模倣するような3テープ多記号可逆的Turing機械Rが存在する。

【命題2．5】(5)　任意の（一般に非可逆な）Turing機械Tに対し、それを模倣するような1テープ2記号可逆的Turing機械Rが存在する。

可逆的Turing機械と可逆的ＰＣＡの関係については次の結果が得られている。

【命題2．6】(6)　任意の1テープ2記号可逆的Turing機械Rに対し、それを模倣するような可逆的3ＰＣＡ，Ｐが存在する。

命題2.2，2.5，2.6より、次の結果が得られる。

【命題2．7】(6)　可逆的3ＰＣＡおよび、可逆的3ＣＡは計算万能である。

3.　2ＰＣＡによる3ＰＣＡの模倣と、可逆的2ＰＣＡの計算万能性

【定理3．1】　任意の3ＰＣＡ，P_3に対し、その動作を模倣する2ＰＣＡ，P_2が存在する。

（証明）　与えられたP_3を

$$P_3=(Z,\ (L,C,R),\ g,\ (l_0,c_0,r_0))$$

とするとき、それを模倣する2ＰＣＡ，P_2は次のようにして構成できる。

$$P_2=(Z,\ (X,Y),\ g',\ (x_0,y_0))$$

$$X\ =(L\times C-\{(l_0,c_0)\})\cup(L-\{l_0\})\cup\{x_0\}$$

$$Y\ =(R-\{r_0\})\cup(R\times C-\{(r_0,c_0)\})\cup\{y_0\}$$

局所写像g'は次のように定める。

（1）任意の

$r \in (\mathrm{R} - \{r_0\})$

$\ell \in (\mathrm{L} - \{\ell_0\})$

$c \in (\mathrm{C} - \{c_0\})$

に対して、g'を次のように定める。

(1a) $g'(r, (\ell, c)) = (\ell, (r, c))$

(1b) $g'(r, (\ell, c_0)) = (\ell, (r, c_0))$

(1c) $g'(r, (\ell_0, c)) = (x_0, (r, c))$

(1d) $g'(r, x_0) = (x_0, (r, c_0))$

(1e) $g'(y_0, (\ell, c)) = (\ell, (r_0, c))$

(1f) $g'(y_0, (\ell, c_0)) = (\ell, y_0)$

(1g) $g'(y_0, (\ell_0, c)) = (x_0, (r_0, c))$

(1h) $g'(y_0, x_0) = (x_0, y_0)$

（2）任意の

$(r_1, c_1) \in (\mathrm{R} \times \mathrm{C} - \{(r_0, c_0)\})$

$\ell_1 \in (\mathrm{L} - \{\ell_0\})$

$(\ell_2, c_2) \in (\mathrm{L} \times \mathrm{C} - \{(\ell_0, c_0)\})$

$r_2 \in (\mathrm{R} - \{r_0\})$

に対して、g'を次のように定める。

(2a) $g(r_1, c_1, \ell_1) = (\ell_2, c_2, r_2)$ ならば

$g'((r_1, c_1), \ell_1) = ((\ell_2, c_2), r_2)$

(2b) $g(r_1, c_1, \ell_1) = (\ell_2, c_2, r_0)$ ならば

$g'((r_1, c_1), \ell_1) = ((\ell_2, c_2), y_0)$

(2c) $g(r_1, c_1, \ell_1) = (\ell_0, c_0, r_2)$ ならば

$g'((r_1, c_1), \ell_1) = (x_0, r_2)$

(2d) $g(r_1, c_1, \ell_1) = (\ell_0, c_0, r_0)$ ならば

$g'((r_1, c_1), \ell_1) = (x_0, y_0)$

(2e) $g(r_1, c_1, \ell_0) = (\ell_2, c_2, r_2)$ ならば

$g'((r_1, c_1), x_0) = ((\ell_2, c_2), r_2)$

(2f) $g(r_1, c_1, \ell_0) = (\ell_2, c_2, r_0)$ ならば

$g'((r_1, c_1), x_0) = ((\ell_2, c_2), y_0)$

(2g) $g(r_1, c_1, \ell_0) = (\ell_0, c_0, r_2)$ ならば

$g'((r_1, c_1), x_0) = (x_0, r_2)$

(2h) $g(r_1, c_1, l_0) = (l_0, c_0, r_0)$ ならば

$$g'((r_1, c_1), x_0) = (x_0, y_0)$$

(2i) $g(r_0, c_0, l_1) = (l_2, c_2, r_2)$ ならば

$$g'(y_0, l_1) = ((l_2, c_2), r_2)$$

(2j) $g(r_0, c_0, l_1) = (l_2, c_2, r_0)$ ならば

$$g'(y_0, l_1) = ((l_2, c_2), y_0)$$

(2k) $g(r_0, c_0, l_1) = (l_0, c_0, r_2)$ ならば

$$g'(y_0, l_1) = (x_0, r_2)$$

(2l) $g(r_0, c_0, l_1) = (l_0, c_0, r_0)$ ならば

$$g'(y_0, l_1) = (x_0, y_0)$$

（３）任意の

$r \in (R - \{r_0\})$

$l \in (L - \{l_0\})$

$(r_1, c_1) \in (R \times C - \{(r_0, c_0)\})$

$(l_1, c_2) \in (L \times C - \{(l_0, c_0)\})$

に対して、g'を次のように定める。

(3a) $g'(r, l) = (l, r)$

(3b) $g'((r_1, c_1), (l_1, c_2)) = ((l_1, c_2), (r_1, c_1))$

P_2は図３．１のように、P_3の１ステップを２ステップかかって模倣する。P_2は第１のステップでP_3の各セルが状態遷移をするのに必要な情報を適当な場所に集め、第２のステップで実際の状態遷移を模倣する。P_2の各セルは、第１のステップの初め（図３．１の時刻ｔ）では、Ｘの部分によってP_3のＬとＣの部分の状態を、Ｙの部分によってＲの部分の状態を保持する。第２のステップの初め（図３．１の時刻ｔ＋１）では、Ｘの部分によってＬの部分の状態を、Ｙの部分によってＲとＣの部分の状態を保持する。また、P_2の静止状態は(x_0, y_0)であるが、x_0は(l_0, c_0)とl_0に、y_0はr_0と(r_0, c_0)に（多義的に）対応しており、各時点でそのいずれを表しているかは、その時点でどちらのステップを実行する段階にあるかで決まる。

上述の規則(1a)-(1h)は第１のステップ、(2a)-(2l)は第２のステップを実行するためのものである。それらの内で、(1b)-(1h),(2b)-(2l)は、静止状態を適切に扱うためのものである。従って、静止状態を持たないようなシステムではこれらの規則は不要である。また、規則(3a),(3b)は模倣には不必要なものなので右辺の値はどのように定めてもよいが、このようにしておくと後述の定理３．２で可逆性

の問題を扱うときに都合がよい。

l_0, c_0, r_0に関係しないセルの状態遷移をP_2が正しく行っていることは規則(1a)と(2a)から容易にわかるが、そうでないセルに対しては場合分けをして個別に確かめる必要がある。場合の数は多いが、それぞれの確認は容易であるのでここでは省略する。 ■

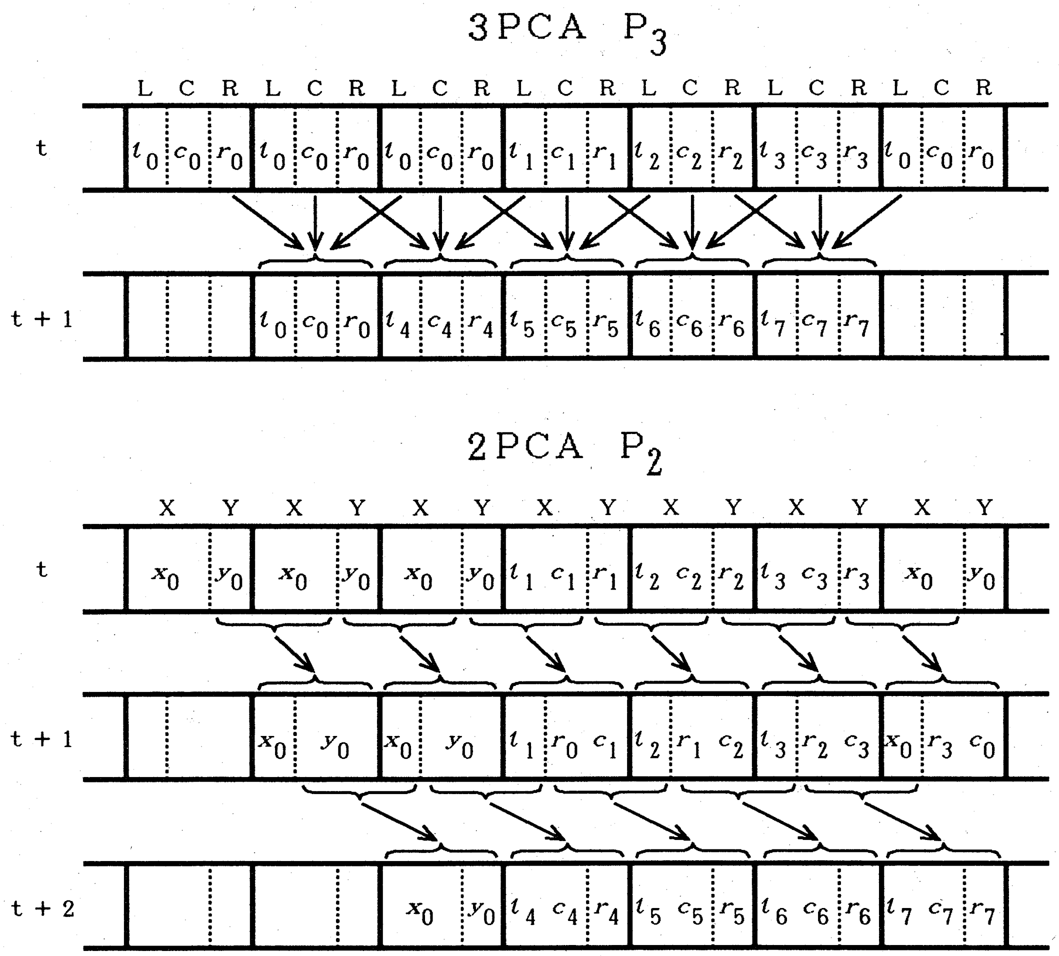

図3．1　2PCA, P_2による3PCA, P_3の模倣

【定理3．2】　任意の可逆的3PCA, P_3に対し、その動作を模倣する可逆的2PCA, P_2が存在する。

（証明）定理3．1の証明における2PCAの構成法は可逆性を保存する、つまり、P_3の局所写像gが単射ならP_2の局所写像g'も単射になることが言える。これは構成法を丹念に調べることによって確かめられるが、その概略は次の通りである。まず規則(1a)〜(1h)は、基本的には、状態l, c, rを保持する位置をP_2の

セルの中で入れ換える操作をしているだけなので、これら規則の範囲内ではg'の単射性が保証される。実際、(1a)～(1h)の中には右辺が一致するような規則は存在しない。規則(2a)～(2l)においても、gが単射ならg'の単射性は崩れないことがわかる。例えば、(1h),(2d),(2h),(2l)のg'の右辺はすべて(x_0, y_0)で一致しているが、gが単射なのでそれらの内の高々1つ（実際には(1h)の場合だけ）しか生じない。g'の右辺が見かけ上一致する他の規則についても同様である。また、規則(3a),(3b)についてもそれらの右辺が一致する規則は他にないことが確かめられるので、これを加えてもg'の単射性は崩れない。 ■

定理３．２と命題２．６より系３．３が、系３．３と命題２．３より系３．４が、それぞれ導かれる。

【系３．３】 可逆的２ＰＣＡは計算万能である。

【系３．４】 可逆的２ＣＡは計算万能である。

参考文献

(1) C.H.Bennett: "Logical reversibility of computation", IBM J. Res. & Dev., **6**, 11, pp.525-532 (1973).

(2) C.F.Bennett: "The thermodynamics of computation", Int.J.of Theoretical Physics, **21**, 12, pp.905-940 (1982).

(3) C.H.Bennett: "Notes on the history of reversible computation", IBM J. Res.& Dev., **32**, 1, pp.16-23 (1988).

(4) E.Fredkin and T.Toffoli: "Conservative logic", Int.J.of Theoretical Physics, 21, 3/4, pp.219-253 (1982).

(5) K.Morita, A.Shirasaki, and Y.Gono: "A 1-tape 2-symbol reversible Turing machine", Trans. IEICE Japan, E72, No.3, pp.223-228 (1989).

(6) K.Morita, and M.Harao: "Computation universality of one-dimensional reversible (injective) cellular automata", Trans. IEICE Japan, E72, No.6, pp.758-762 (1989).

(7) T.Toffoli: "Computation and construction universality of reversible cellular automata", J.Comput. & Syst.Sci., **15**, pp.213-231 (1977).